7T

УДК 004.81

В.Ю. Тітова
Хмельницький національний університет МОН України, м. Хмельницький
Україна, 29016, м. Хмельницький, вул. Інститутська, 11

Нейромережний метод для визначення якості програмного забезпечення критичного застосування

V.Y. Titova
Khmelnitsky National University MES of Ukraine, c. Khmelnitsky
Ukraine, 29016, t. Khmelnitsky, Instytutska st., 11

Neural Network Method for Determining the Quality of Critical Software

В.Ю. Титова
Хмельницкий национальный университет МОН Украины, г. Хмельницкий
Украина, 29016, г. Хмельницкий, ул. Институтская, 11

Нейросетевой метод для определения качества критического программного обеспечения

У статті розглянуто задачу оцінювання якості програмного забезпечення критичного застосування. Визначено, що існуюча на сьогоднішній день модель якості є неформалізованою. А тому авторами було запропоновано формалізовану модель якості програмного забезпечення критичного застосування.
Ключові слова: програмне забезпечення критичного застосування, модель якості, характеристики якості, формалізована модель, нейронна мережа.

The problem of evaluation of critical software quality has been examined in the article. It is determined that the current model of quality is non-formalized. Therefore, the authors proposed the formalized model of quality of critical software.
Key words: critical software, quality model, quality specifications, formalized model, neural network.

В статье рассмотрена задача оценивания качества критичного программного обеспечения. Определено, что существующая на сегодняшний день модель качества является неформализованной. А потому авторами была предложена формализованная модель качества программного обеспечения критического применения.
Ключевые слова: критическое программное обеспечение, модель качества, характеристики качества, формализованная модель, нейронная сеть.

Вступ

Успішний розвиток більшості сучасних галузей в Україні багато в чому залежить від досконалості інформаційних технологій. Інформаційні технології, ядром яких є програмне забезпечення (ПЗ) критичного застосування, застосовуються у автоматизованих системах керування у космічній та авіаційній галузі; атомній галузі; медицині та інших сферах.

А тому надійність і безпека роботи космічних апаратів, літаків, атомних станцій, медичного обладнання прямо залежать від якості ПЗ. Необхідною умовою підвищення якості ПЗ критичного застосування є розроблення нормативної бази, що визначає вимоги до якості зазначеного ПЗ, та методи оцінювання реалізації цих вимог.

Характеристика предметної області

ПЗ критичного застосування – це ПЗ, що виконує критичні функції, важливі для безпеки, тобто ПЗ, відмова у виконанні функцій якого (втрати або деградації) чи його неправильна або недбала експлуатація можуть призвести до катастрофічних або критичних наслідків. Наприклад, це ПЗ, яке:

– безпосередньо виконує команди і контроль за умовами або станом апаратних компонентів, і його невиконання, непослідовне виконання або неправильне виконання може спричинити неправильне виконання функцій керування, що наражає на ризик безпеки або може призвести до виникнення небезпечних умов;

– здійснює поточний контроль за станом апаратних компонентів, і результатами його невиконання, непослідовного виконання або неправильного виконання можуть стати дані, що призводять до помилкових рішень операторів або допоміжних систем, що наражає на ризик безпеки або може призвести до існування умов виникнення ризиків;

– безпосередньо виконує управління і контроль за умовами або станом апаратних компонентів, і його невиконання, непослідовне виконання або виконання, пов'язане з помилками персоналу, апаратною відмовою або відмовою внаслідок впливу навколишнього середовища може спричинити небезпеку або може призвести до існування умов виникнення ризиків [1].

Якість ПЗ повинна оцінюватися з використанням моделі якості. Модель якості визначають після розроблення вимог до якості готового ПЗ та проміжного продукту і вона є сукупністю атрибутів якості ПЗ, класифікованих в ієрархічну деревоподібну структуру характеристик і підхарактеристик. Верхній рівень цієї структури складається з характеристик якості, а нижній – з атрибутів якості ПЗ. Ця ієрархія не є строгою, тому що деякі атрибути можуть входити більш ніж в одну характеристику [2].

Відповідно до стандарту [2] модель якості ПЗ складається із двох частин:

– внутрішня і зовнішня якість;

– якість у використанні.

Модель внутрішньої та зовнішньої якості визначає шість основних характеристик якості ПЗ [3], [4]:

– функціональність (Functionality) – здатність ПЗ забезпечити функції, які виконують заявлені потреби та потреби, що маються на увазі при використанні ПЗ в заданих умовах, включає наступні підхарактеристики: функціональна повнота (Suitability), правильність (Accuracy), здатність до взаємодії (Interoperability), захищеність (Security), узгодженість функціональності (Functionality Compliance);

– надійність (Reliability) – група властивостей, що обумовлює здатність програмного забезпечення зберігати працездатність та перетворювати вихідні дані в очікуваний результат у заданих умовах за встановлений час, включає наступні підхарактеристики: безвідмовність (Maturity), стійкість до відхилень (Fault Tolerance), відновлюваність (Recoverability), узгодженість надійності (Reliability Compliance);

– зручність використання (Usability) – здатність ПЗ бути зрозумілим, придатним до вивчання і привабливим для користувача при його використанні в заданих умовах, включає наступні підхарактеристики: зрозумілість (Understandability), придатність до вивчання, зручність інтерфейсу для користування (Operability), привабливість (Attractiveness), узгодженість використання (Usability Compliance);

– раціональність (Efficiency) – здатність ПЗ забезпечувати відповідну (допустиму) продуктивність з урахуванням займаних ресурсів у заданих умовах, включає наступні підхарактеристики: часова раціональність (Time Behavior), використовуваність ресурсів (Resource Utilization), узгодженість раціональності (Efficiency Compliance);

– супроводжуваність (Maintainability) – здатність ПЗ до модифікації, що може містити у собі виправлення, поліпшення або адаптацію ПЗ до змін середовища, вимог або функціональних специфікацій, включає наступні підхарактеристики: аналізованість (Analyzability), змінюваність (Changeability), стабільність (Stability), тестопридатність (Testability), узгодженість супроводжуваності (Maintainability Compliance);

– переношуваність (Portability) – здатність ПЗ бути перенесеним з одного організаційного, апаратного або програмного середовища в інше, включає наступні підхарактеристики: адаптованість (Adaptability), налагоджуваність (Installability), сумісність або безконфліктність (Coexistence), взаємозамінність (Replaceability), узгодженість переношуваності (Portability Compliance).

Модель якості у використанні визначає наступні чотири характеристики [5]:

– ефективність (Effectiveness) – здатність програмного забезпечення дозволяти користувачеві досягати зазначених цілей з точністю та повнотою в заданому контексті використання;

– продуктивність (Productivity) – здатність програмного забезпечення дозволяти користувачеві витрачати відповідну кількість ресурсів щодо ефективності, досягнутої в заданому контексті використання;

– безпека (Safety) – здатність програмного забезпечення досягати прийнятних рівнів ризику шкоди людям, бізнесу, ПЗ, майну або навколишньому середовищу в заданому контексті використання;

– задовільність (Satisfaction) – здатність програмного забезпечення задовольняти користувачів у заданому контексті використання.

Характеристики та підхарактеристики є підставою для розроблення вимог до якості ПЗ. Модель якості дає можливість визначити якість ПЗ та оцінити його у процесах життєвого циклу, що включають процеси придбання, формування вимог, розроблення, використання, супроводу, забезпечення якості та аудиту, і може використовуватися розробником, покупцем, службою забезпечення якості, незалежними експертами.

Постановка задачі

Характеристики якості ПЗ, наведені вище, поділяються на [3-5]:

– описові, які описують набір засобів і загальні характеристики об'єкта, його функції, безпеку, захищеність і важливість;

– кількісні, які можна виміряти та чисельно зіставити з вимогами;

– якісні, які визначають експертним методом.

А тому можна зробити висновки, що через різнотипність вхідних даних для вирішення задачі оцінювання якості ПЗ критичного застосування не можливо застосувати математичні методи, що ускладнює розрахунок якості ПЗ.

Формалізація моделі якості ПЗ дозволила б спростити вирішення вищевказаної задачі та, як результат, підвищити якість використовуваного ПЗ.

Формалізована модель якості ПЗ критичного застосування

Для формалізації моделі використаємо апарат теорії множин [6].

Позначимо загальну якість ПЗ – Q. Вона обчислюється зі значень характеристик множин внутрішньої і зовнішньої якості A та якості у використанні B.

В свою чергу показник A визначається з множини характеристик функціональності F, $F=[f_1,f_2,f_3,f_4]$, де f_1 – функціональна повнота, f_2 – правильність, f_3 – здатність до взаємодії, f_4 – захищеність; надійності R, $R=[r_1,r_2,r_3]$, де r_1 – безвідмовність, r_2 – стійкість до відхилень, r_3 – відновлюваність; зручності використання U, $U=[u_1,u_2,u_3,u_4]$, де u_1 – зрозумілість, u_2 – придатність до вивчання, u_3 – зручність інтерфейсу для користування, u_4 – привабливість; раціональності E, $E=[e_1,e_2]$, де e_1 – часова раціональність, e_2 – використовуваність ресурсів; супроводжуваності M, $M=[m_1,m_2,m_3,m_4]$, де m_1 – аналізованість, m_2 – змінюваність, m_3 – стабільність, m_4 – тестопридатність; переношуваності P, $P=[p_1,p_2,p_3,p_4]$, де p_1 – адаптованість, p_2 – налагоджуваність, p_3 – взаємозамінність, p_4 – сумісність; узгодженості C, $C=[c_1,c_2,c_3,c_4,c_5,c_6]$, де c_1 – узгодженість функціональності, c_2 – узгодженість надійності, c_3 – узгодженість використання, c_4 – узгодженість раціональності, c_5 – узгодженість супроводжуваності, c_6 – узгодженість переношуваності.

Оскільки ці характеристики є взаємонезалежними, для визначення зовнішньої та внутрішньої якості доцільно буде використати адитивний показник або операцію об'єднання для множин. Отже:

$$A=F\cup R\cup U\cup E\cup M\cup P\cup C. \quad (1)$$

Множина характеристик якості використання B складається з атрибутів ефективності ef, продуктивності pr, безпеки sf та задовільності st.

$$B=\{ef,pr,sf,st\}. \quad (2)$$

І ці атрибути є залежними від характеристики зовнішньої та внутрішньої якості. Так, наприклад, ефективність залежить від характеристик функціональності, надійності, супроводжуваності, переношуваності, продуктивність – від характеристик раціональності, безпека – від характеристик надійності, задовільність – від характеристик функціональності та зручності використання.

А тому для знаходження показника загальної якості використаємо мультиплікативний показник або добуток множин.

$$Q=A\times B. \quad (3)$$

Тому формалізовану модель якості програмного забезпечення можна представити у наступному вигляді $Q=(F\cup R\cup U\cup E\cup M\cup P\cup C)\times B$. Якщо замінити множини відповідними їм характеристиками та відкрити дужки, то отримаємо формулу (4), яка і є формалізованою моделлю якості програмного забезпечення.

З аналізу отриманої формалізованої моделі можна зробити наступні висновки:

– задачу оцінювання якості ПЗ критичного застосування неможливо розв'язати шляхом повного перебору всіх наявних варіантів, через велику кількість можливих рішень та ще більшу кількість близьких між собою характеристик;

– дану задачу неможливо вирішити шляхом застосування методів розв'язку систем алгебраїчних рівнянь через обмеженість комп'ютерних ресурсів та тому, що розв'язок даної задачі не може бути зведений до звичайних числових розрахунків.

$$Q=\left\{\begin{array}{l}(f_1,ef),(f_2,ef),(f_3,ef),(f_4,ef),(r_1,ef),(r_2,ef),\\(r_3,ef),(u_1,ef),(u_2,ef),(u_3,ef),(u_4,ef),(e_1,ef),\\(e_2,ef),(m_1,ef),(m_2,ef),(m_3,ef),(m_4,ef),(p_1,ef),\\(p_2,ef),(p_3,ef),(p_4,ef),(c_1,ef),(c_2,ef),(c_3,ef),\\(c_4,ef),(c_5,ef),(c_6,ef),(f_1,pr),(f_2,pr),(f_3,pr),\\(f_4,pr),(r_1,pr),(r_2,pr),(r_3,pr),(u_1,pr),(u_2,pr),\\(u_3,pr),(u_4,pr),(e_1,pr),(e_2,pr),(m_1,pr),(m_2,pr),\\(m_3,pr),(m_4,pr),(p_1,pr),(p_2,pr),(p_3,pr),\\(p_4,pr),(c_1,pr),(c_2,pr),(c_3,pr),(c_4,pr),\\(c_5,pr),(c_6,pr),(f_1,sf),(f_2,sf),(f_3,sf),(f_4,sf),\\(r_1,sf),(r_2,sf),(r_3,sf),(u_1,sf),(u_2,sf),(u_3,sf),\\(u_4,sf),(e_1,sf),(e_2,sf),(m_1,sf),(m_2,sf),(m_3,sf),\\(m_4,sf),(p_1,sf),(p_2,sf),(p_3,sf),(p_4,sf),\\(c_1,sf),(c_2,sf),(c_3,sf),(c_4,sf),(c_5,sf),(c_6,sf),(f_1,st),\\(f_2,st),(f_3,st),(f_4,st),(r_1,st),(r_2,st),(r_3,st),(u_1,st),\\(u_2,st),(u_3,st),(u_4,st),(e_1,st),(e_2,st),(m_1,st),(m_2,st),\\(m_3,st),(m_4,st),(p_1,st),(p_2,st),(p_3,st),(p_4,st),(c_1,st),\\(c_2,st),(c_3,st),(c_4,st),(c_5,st),(c_6,st)\end{array}\right\}. \quad (4)$$

Отже, задача оцінювання якості ПЗ критичного застосування належить до важкоформалізованих задач, а тому для її вирішення доцільним буде застосувати методи інтелектуальні, зокрема штучні нейронні мережі. Зазначена нейромережа дозволить на основі характеристик якості провести оцінювання якості ПЗ критичного застосування.

Структура нейромережі для вирішення задачі оцінювання якості ПЗ критичного застосування

Структура нейромережі для вирішення задачі оцінювання якості ПЗ критичного застосування зображена на рис. 1.

Вона має три шари. Кількість входів дорівнює кількості елементів у формалізованій моделі, тобто 108, кількість виходів один.

На нейрони першого шару надходять значення характеристик моделі ситуації.

Нейрони другого шару визначають внутрішню та зовнішню якість, а також якість у використанні.

Нейрон третього шару визначає якість ПЗ критичного застосування.

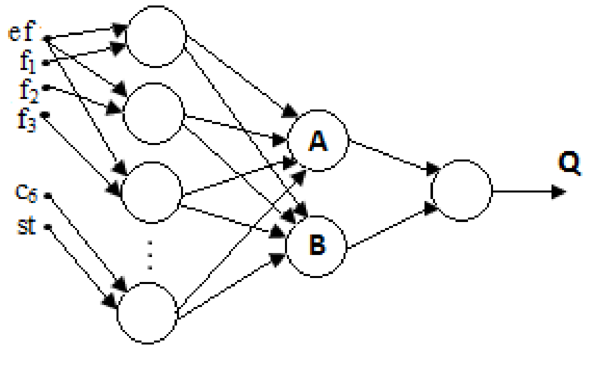


Рисунок 1 – Структура нейромережі для вирішення задачі оцінювання якості ПЗ критичного застосування

Для створення навчальних вибірок з різнотипних вхідних даних нейромережі скористаємося функціями належності з теорії нечітких множин [7].

Для кожного з вхідних параметрів сформуємо шкалу відповідності ПЗ критичного застосування вимогам:

1 – дуже низька відповідність, вимога фактично не дотримується;

3 – низька відповідність, вимога реалізована лише частково;

5 – середня відповідність, вимога реалізована наполовину;

7 – висока відповідність, вимога реалізована практично повністю;

9 – надвисока відповідність, вимога реалізована повністю.

Дані цієї шкали визначаються експертним шляхом, за результатами опитування як розробників, так і замовників та користувачів.

І на основі цієї шкали сформуємо функції належності для зазначених параметрів. На рис. 2 – 3 наведені деякі з них.

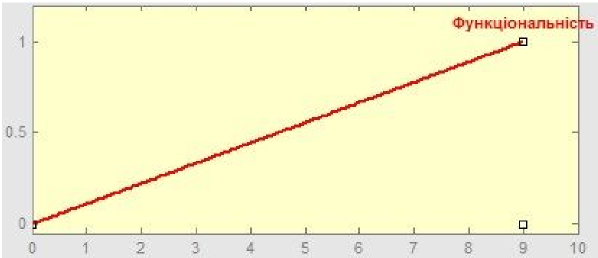


Рисунок 2 – Функція належності параметра функціональності

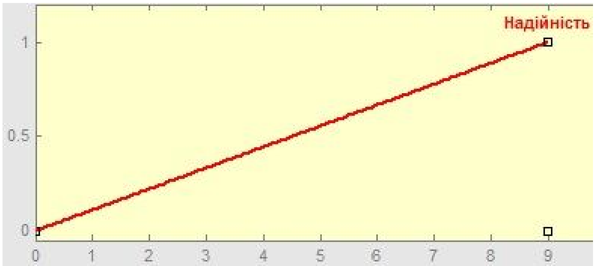


Рисунок 3 – Функція належності параметра надійності

Неповнозв’язну прямонапрямлену нейронну мережу визначення якості ПЗ критичного застосування було побудовано у пакеті Matlab та навчено за допомогою алгоритмів градієнтного спуску *'traingd'*, градієнтного спуску з параметром швидкості настроювання *'traingda'*, градієнтного спуску Моллера *'trainscg'*, оберненого поширення помилки *'trainrp'*, методу січної OSS *'trainoss'* та Левенберга-Марквардта *'trainlm'*. Результати навчання мережі представлені у табл. 1

Таблиця 1 – Результати навчання нейромережі за відповідними алгоритмами

Алгоритм	К-сть епох	Час навчання	Досягнута точність
Градієнтного спуску	1000	1 год 4 хв	0,055
Градієнтного спуску з парам. швидкості настроювання	1000	1 год 1 хв	0,01
Градієнтного спуску Моллера	1000	2 год 8 хв	0,0055
Методу січної OSS	1000	2 год 10 хв	0,005
Оберненого поширення помилки	1000	51 хв 20 с	0,005
Левенберга-Марквардта	1000	16 год 48 хв	0,0055

З аналізу результатів навчання нейромережі слідує, що найкращу точність було досягнуто при використанні алгоритмів градієнтного спуску Моллера, методу січної, оберненого поширення помилки та Левенберга-Марквардта. Найшвидше мережа навчалася за алгоритмами градієнтного спуску, градієнтного спуску з параметром швидкості настроювання та оберненого поширення помилки.

Отже, навчання нейромережі за алгоритмом оберненого поширення помилки дозволяє досягнути найкращої точності за найменший час. Тому для визначення якості ПЗ критичного застосування було використано неповнозв'язну прямонапрямлену нейромережу, навчену за алгоритмом оберненого поширення помилки (рис. 4).

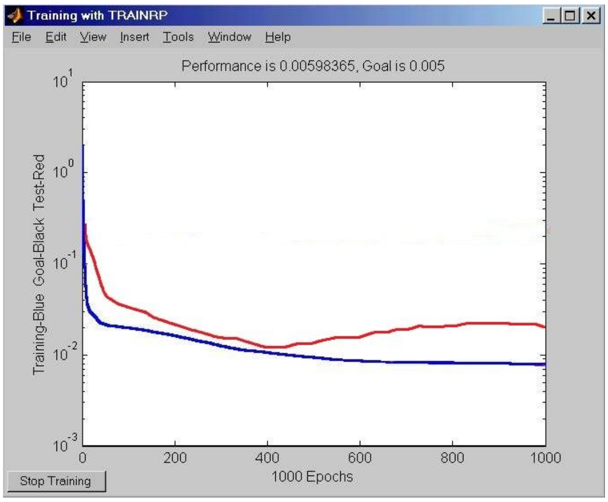


Рисунок 4 – Приклад навчання нейромережі за алгоритмом оберненого поширення помилки

Висновки

У роботі було проведено формалізацію умов, які характеризують модель якості ПЗ критичного застосування. На основі формалізації було розроблено формалізовану модель якості, яка є основою диверсного методу визначення якості ПЗ критичного застосування, та запропоновано структуру нейромережі для її вирішення. Розроблена нейромережа є основою для створення системи оцінювання відповідності ПЗ критичного застосування нефункційним вимогам за результатами тестування.

Література

1. ECSS-Q-80-03 Software Dependability and Safety Methods and Techniques (Методи і методики безпеки та надійності програмного забезпечення).
2. ISO/IEC 9126-1:2001 Software Engineering – Product Quality – Part 1: Quality model (Інженерінг програмного забезпечення – Якість продукту – Частина 1: Модель якості).
3. ISO/IEC TR 9126-2:2003 Software Engineering – Product Quality – Part 2: External metrics (Інженерінг програмного забезпечення – Якість продукту – Частина 2: Зовнішні метрики).
4. ISO/IEC TR 9126-3:2003 Software Engineering – Product Quality – Part 3: Internal metrics (Інженерінг програмного забезпечення – Якість продукту – Частина 3: Внутрішні метрики).
5. ISO/IEC TR 9126-4:2004 Software Engineering – Product Quality – Part 4: Quality in use metric (Інженерінг програмного забезпечення – Якість продукту – Частина 4: Метрики якості у використанні).
6. Верещагин Н.К. Начала теории множеств / Н.К. Верещагин, А. Шень. – М. : МЦНМО, 2002. – 128 с.
7. Круглов В.В. Нечёткая логика и искусственные нейронные сети / Круглов В.В., Дли М.И., Голунов Р.Ю. – М. : Физматлит, 2001. – 221 с.

Literatura

1. ECSS-Q-80-03 Software Dependability and Safety Methods and Techniques.
2. ISO/IEC 9126-1:2001 Software Engineering – Product Quality – Part 1: Quality model.
3. ISO/IEC TR 9126-2:2003 Software Engineering – Product Quality – Part 2: External metrics.
4. ISO/IEC TR 9126-3:2003 Software Engineering – Product Quality – Part 3: Internal metrics.
5. ISO/IEC TR 9126-4:2004 Software Engineering – Product Quality – Part 4: Quality in use metric.
6. Vereshchagin N.K.. Nachala teorii mnodzestv. M.: MCMNO. 2002. 128 s.
7. Kruglov V.V. Nechtkaya logika i neyronnye seti. M.: Fizmatlit. 2001. 221 s.

RESUME

V.Y. Titova

Neural Network Method for Determining the Quality of Critical Software

Critical software is the software that performs critical functions important to safety, i.e. the software, whose failure in performance of functions (loss or degradation) or incorrect operation may lead to catastrophic or critical consequences.

Quality software must be evaluated using the model quality. The model of quality is determined after the development of requirements for quality of the finished software and intermediate and it is a collection of attributes of software quality, classified in a hierarchical tree structure characteristics. The upper level of this structure has the characteristics of quality, and the bottom has the attributes of software quality.

Characteristics are the basis for the development of requirements for software quality. The model of quality can determine the quality of software and evaluate it in the process life cycle, including the processes of acquisition, formation requirements, development, use, maintenance, quality assurance and audit, and can be used by the developer, purchaser, service quality, independent experts.

Formalization of software quality model would allow simplifying the solution of the above problems and improving consequently the quality of the used software.

Стаття надійшла до редакції 05.06.2012.